Improved approximation of the Soft-Capacitated Facility Location Problem

Laurent Alfandari

Improved approximation of the general Soft-Capacitated Facility Location Problem

Laurent Alfandari,
ESSEC,
BP 105 F-95021 Cergy-Pontoise, France *
E-mail : *alfandari@essec.fr*

Abstract

The Soft-Capacitated Facility Location Problem, where each facility is composed of a variable number of fixed-capacity production units, has been recently studied in several papers, especially in the metric case. In this paper, we only consider the general problem where connection costs do not systematically satisfy the triangle inequality property. We show that an adaptation of the set covering greedy heuristic, where the subproblem is approximately solved by a Fully Polynomial-Time Approximation Scheme based on cost scaling and dynamic programming, achieves a logaritmic approximation ratio of $(1+\epsilon)H(n)$ for the problem, where n is the number of customers to be served and H is the harmonic series. This improves the previous bound of $2H(n)$ for this problem.

Key-words: facility location, set covering, dynamic programming, FPTAS.

1 Introduction

The classical single-source Capacitated Facility Location Problem (CFLP) consists in assigning a set of n customers with known demands to a set of m possible facilities so that each customer is assigned to a single facility without violating capacities of open facilities, while minimizing the sum of the construction cost of selected facilites and the connection cost of customers to facilities. In this paper, we consider a variant of CFLP where each facility, if open, can be composed of a variable number (to determine) of fixed-size production units. This problem, known as the Soft-Capacitated Facility Location Problem (SCFLP), was first introduced in [11]. It arises indeed in many industrial applications, as production is often structured by production lines or teams whose number is a decision to make. For large instances of hard problems, the design of heuristics that are both fast and efficient is a challenge. In this field, the polynomial approximation theory has received much attention in the last two decades. The aim is to develop a ρ-approximation of the problem, i.e., a polynomial-time algorithm that finds a feasible solution whose objective function is always within a factor ρ of the optimum, so that ρ is as small as possible. The best-known approximation ratio for

*also LIPN, CNRS UMR-7030, Université Paris XIII, France the metric version of CFLP is $6(1+\epsilon)$ and was produced by Zhang, Chen and Ye [20], then by Garg, Khandekar and Pandit [7] for a more general version of the problem, using a local search algorithm. This result generalizes the one previously found by Chudak and Williamson [3] for the case when all capacities are the same. The first constant approximation ratio for the metric uncapacitated problem (UFLP) was found by Shmoys, Tardos and Aardal [19]. Their method, achieving an approximation ratio of 3.16, is based on LP-rounding. This ratio has been repeatedly improved then until the greedy algorithm of Mahdian, Ye and Zhang [17] which provides an approximation ratio of 1.52 for UFLP. The metric version of SCFLP was shown by Jain, Mahdian and Saberi to admit a 3-approximation by a combination of a primal-dual greedy process and lagrangian relaxation [13]. This ratio was recently improved by the same authors to a 2-approximation [18]. However, the metric case is not general enough to capture such a natural setting as connection costs depending on the quantity of demand transported. For example, let us assume that connection costs c_{ij} from a client i to a facility j are transportation costs which are linear in the distance in kilometers δ_{ij} and the quantity d_i (say, in tons) delivered to the client, i.e., $c_{ij} = \delta_{ij} d_i \mu$, where μ is a unitary transportation cost expressed in currency units per kilometer and ton. Consider then two facilities j and j' and two customers i and i' such that $\delta_{ij} = \delta_{i'j'} = 50$, $\delta_{ij'} = \delta_{i'j} = 30$, $d_i = 1000$ and $d_{i'} = 100$. We have but $c_{ij} = 50000\mu$ and $c_{ij'} + c_{i'j'} + c_{i'j} = 38000\mu$, so the triangle inequality $c_{ij} \leq c_{ij'} + c_{i'j'} + c_{i'j}$ does not hold. Therefore, approximating the general (non-metric) problem is a real issue. The general SCFLP is approximable within ratio $2H(n)$, where $H(n) = \sum_{1 \leq i \leq n} 1/i$ (see [11]). This comes from the fact that a ρ-approximation for UFLP provides a 2ρ-approximation for SCFLP, and UFLP was shown to be approximable within ratio $H(n)$ by Hochbaum [9]. The algorithm of [9] for UFLP relies on an exponential-size Set Covering reformulation of UFLP and the fact that the exponential set of candidate subsets can be reduced to an equivalent set of polynomial size. Since the Set Covering Problem (SCP) is approximable within ratio $H(n) \leq 1 + \ln n$ [4], the result also holds for UFLP. The bound of $O(\ln n)$ is asymptotically tight for SCFLP since the problem is linked by an approximation-preserving reduction with SCP and SCP cannot be approximated within a ratio better than $\ln n - \ln \ln n$ [5]. We improve the ratio of $2H(n)$ for SCFLP to $(1+\epsilon)H(n)$ by an algorithm running in time $O(mn^4/\epsilon)$. This algorithm also uses an exponential-size Set Covering reformulation of SCFLP, and a FPTAS based on cost scaling and rounding and dynamic programming for the subproblem of the SCP greedy heuristic. In our approach, we do not restrict a priori the collection of subsets in the SCP reformulation and do not exactly solve the subproblem, contrary to the approach developped in [1].

The SCFLP is formally stated and reformulated as a SCP in section 2. The adaptation of the SCP classical greedy process to SCFLP is presented in section 3. The subproblem of the greedy heuristic for SCFLP is shown to admit a Fully Polynomial Time Approximation Scheme (FPTAS) in section 4. Section 5 concludes the paper.

2 Problem statement and reformulation

The Soft-Capacitated Facility Location Problem (SCFLP) is stated as follows. The set of customers to be served is denoted by $I = \{1, \ldots, n\}$, whereas the set of possible locations for facilities is $J = \{1, \ldots, m\}$. For $(i, j) \in I \times J$, c_{ij} is the connection cost between customer i and location j, d_i is the demand of customer i, f_j (resp. u_j) is the construction cost (resp., capacity) of a production line on location j. The Integer Linear Programming model corresponding to SCFLP is the following:

$$\begin{aligned}
& & \text{Minimize } & \sum_{j\in J} f_j y_j + \sum_{(i,j)\in I\times J} c_{ij}x_{ij} & & (1)\\
& \text{s.t.} & & \sum_{j\in J} x_{ij} = 1 & \text{for } i \in I & \quad (2)\\
& \text{(SCFLP)} & & \sum_{i\in I} d_i x_{ij} \le u_j y_j & \text{for } j \in J & \quad (3)\\
& & & y_j \in \mathbf{N}, x_{ij} \in \{0,1\} & & (4)
\end{aligned}$$

where integer variables y_j indicate the number of production lines settled in facility $j \in J$, and binary variables x_{ij} indicate whether customer $i \in I$ is assigned to location $j \in J$ or not. The objective (1) minimizes the total cost of the location. The semi-assignment constraints (2) express single-source supplying. Constraints (3) express restricted capacities of facilities. The difference between SCFLP and the classical CFLP is that variables y_j are not binary but integer (and unbounded). SCFLP is NP-hard, since the Set Covering Problem (SCP), which is NP-hard [6], reduces to it. Given a set C of elements and a collection $\mathcal{S} = \{S_1, \ldots, S_m\}$ of subsets of C with cost $c(S)$ for $S \in \mathcal{S}$, SCP consists in finding a minimum cover of C, i.e., a subset $\mathcal{S}' \subseteq \mathcal{S}$ such that $\cup_{S\in\mathcal{S}'} S = C$ and total cost $\sum_{S\in\mathcal{S}'} c(S)$ is minimum. The polynomial reduction is built as follows: set $I = C$, $J = \mathcal{S}$, $u_j = n$, $f_j = c(S_j)$ for all $j \in J$, $d_i = 1$ for all $i \in I$, and $c_{ij} = 0$ if $i \in S_j$, M otherwise, with $M > \sum_{j\in J} f_j$. Then, there is a SCFLP solution of cost at most c if and only if there is a cover of cost at most c in the transformed set covering instance.

The best-known ratio for the general (non-metric) SCFLP relies on a reduction to the uncapacitated problem UFLP. The formulation of UFLP is: minimize (1) under constraints (2) and $y_j \ge x_{ij}$ for all $i, j \in I \times J$, where variables y_j are binary. The result mentionned in section 1, according to which a ρ-approximation for UFLP provides a 2ρ-approximation for SCFLP [11], is obtained by replacing connection costs c_{ij} by $c_{ij} + d_i(f_j/u_j)$ in UFLP. The approximation result of $2H(n)$ for SCFLP is achieved by applying Hochbaum's approach to UFLP with the modified connection costs. Our improvement of this bound is achieved by reformulating SCFLP as a particular SCP. The key idea is that approximately solving the subproblem of the exact SCFLP problem reveals to be better than exactly solving the subproblem of the approximate UFLP model. We introduce now the SCP reformulation of SCFLP.

Definition 1. Let $\mathcal{I}$ be an arbitrary instance of SCFLP. We denote by $\gamma(\mathcal{I})$ the transformed Set Covering instance of $\mathcal{I}$ such that:

(i) $C = I$ is the set of elements to cover,

(ii) $\mathcal{S} = \{S_{L,j} :\ L \subseteq I, j \in J\}$ is the collection of subsets,

(iii) each subset $S_{L,j} \in \mathcal{S}$ covers L and has cost $c(S_{L,j}) = \lceil \sum_{i\in L} d_i/u_j \rceil f_j + \sum_{i\in L} c_{ij}$

Proposition 1. *Solving SCFLP on an arbitrary instance $\mathcal{I}$ is equivalent to solve SCP on $\gamma(\mathcal{I})$, i.e., every SCFLP-solution of cost at most c for $\mathcal{I}$ can be transformed in polynomial* *time in a cover of cost at most c for $\gamma(\mathcal{I})$.*

Proof. Let $\{y_j, x_{ij}\}$ be a solution of VFCLP on $\mathcal{I}$ with cost c. Then, the collection of subsets $\{S_{L(j),j} : j \in J / y_j > 0\}$, where $L(j) = \{i \in I : x_{ij} = 1\}$, is a feasible cover in $\gamma(\mathcal{I})$. From (3) and (4) we have $\lceil \sum_{i\in I} d_i x_{ij}/u_j \rceil \leq y_j$ and we easily derive that the cost of the cover is at most c. Conversely, let $\mathcal{S}' = \{S_{L^t,j^t}, t = 1, \ldots, q\} \subset \mathcal{S}$ be a feasible cover for $\gamma(\mathcal{I})$. Set $Q^1 = L^1$ and $Q^t = L^t \setminus \cup_{1\leq h\leq t-1} L^h$ for $t = 2, \ldots, q$. Set $x_{ij^t} = 1$ for all $i \in Q^t$, $t = 1, \ldots, q$, set all other x-variables to zero, and $y_j = \lceil \sum_{i\in I} d_i x_{ij}/u_j \rceil$ for $j \in J$. This solution satisfies (2-4) and thus is indeed a feasible solution of SCFLP. We get

$$\begin{aligned}
\sum_{j\in J} f_j y_j + \sum_{(i,j)\in I\times J} c_{ij}x_{ij} &\leq \sum_{t=1}^{q}\left(\lceil \frac{\sum_{i\in Q^t} d_i}{K_{j^t}} \rceil f_{j^t} + \sum_{i\in Q^t} c_{ij}\right) \text{ as } y_j \leq \sum_{t:j^t=j} \lceil \frac{\sum_{i\in Q^t} d_i}{K_{j^t}} \rceil \\
&\leq \sum_{t=1}^{q}\left(\lceil \frac{\sum_{i\in L^t} d_i}{K_{j^t}} \rceil f_{j^t} + \sum_{i\in L^t} c_{ij}\right) \text{ as } Q^t \subseteq L^t \\
&= c(\mathcal{S}')
\end{aligned}$$

which completes the proof. □.

3 Greedy heuristic and worst-case analysis

Since SCFLP reduces to SCP by proposition 1, we consider the best polynomial-time algorithm for SCP, i.e., the *Greedy* heuristic which picks at each step a subset $S^* \in \mathcal{S}$ minimizing the ratio 'cost over number of new covered elements'. If U denotes the set of elements that remains to cover at current step, the *subproblem* of the *Greedy* heuristic is formally described as finding

$$r^*(U) = \min_{S\in\mathcal{S}} \frac{c(S)}{|S\cap U|} \tag{5}$$

This iterative search terminates when $U = \emptyset$. The *Greedy* heuristic was shown by Chvátal to guarantee an approximation ratio of $H(\Delta) \leq 1 + \ln \Delta$, where $\Delta = \max_{S\in\mathcal{S}} |S|$ [4]. Nevertheless, this heuristic cannot be directly applied to the SCP instance $\gamma(\mathcal{I})$, given an instance $\mathcal{I}$ of SCFLP, since the number $|\mathcal{S}|$ of candidate subsets in $\gamma(\mathcal{I})$ is equal to $|J|2^{|I|} = m2^n$, which is exponential in n (hence the reduction of definition 1 is not a polynomial Karp-reduction [15]). Therefore, enumeration of $\mathcal{S}$ for solving subproblem (5) is prohibited. We first use the fact that that if subproblem (5) is approximable within ratio $(1+\epsilon)$ then the logaritmic approximation ratio of *Greedy* is conserved (Proposition 2). Then we prove that the subproblem for SCFLP, which is NP-hard (Proposition 3), admits indeed a polynomial-time $(1+\epsilon)$-approximation despite the exponential number of subsets in $\gamma(\mathcal{I})$ (Proposition 4). For proposition 2, we need the following lemma that reformulates for our needs a part of the proof of [4].

Lemma 1. [4] *Let $\mathcal{S}' = \{S_1, \ldots, S_q\}$ be a feasible cover of C for SCP. For $S \in \mathcal{S}$, let $S^1 = S$ and $S^t = S \setminus \cup_{1\leq h\leq t-1} S_h$ for $t = 2, \ldots, q$. Moreover, set $t_S = \max\{t : S^t \neq \emptyset\}$.*

Then we have

$$c(\mathcal{S}') \leq \sum_{S \in \mathcal{S}^{opt}} \left(\sum_{t=1}^{t_S} \left(|S^t| - |S^{t+1}|\right) \left(\frac{c(S_t)}{|S_t^t|}\right) \right) \tag{6}$$

where S^{opt} is an optimal cover.

Proposition 2. *Consider an instance $(C, \mathcal{S})$ of the Set Covering problem. If the subproblem (5) can be approximated within ratio $1+\epsilon$ by some polynomial-time algorithm A, then the associated greedy heuristic $Greedy(A)$, where A is applied to the subproblem, approximates the Set Covering instance within ratio $(1+\epsilon)H(\Delta)$, where $\Delta = \max_{S \in \mathcal{S}} |S|$.*

Proof. The proof simply adapts Chvátal's one. Let $\mathcal{S}' = \{S_1, \ldots, S_q\}$ denote the cover constructed by $Greedy(A)$ in chronological order $1, \ldots, q$. Since A is a $(1+\epsilon)$ approximation for the subproblem, the subset S_t^t defined as in lemma 1 satisfies $c(S_t)/|S_t^t| \leq (1+\epsilon)(c(S)/|S^t|)$ for all $S \in \mathcal{S}$. Plugging that inequality into (6) leads to

$$\begin{aligned} c(\mathcal{S}') &\leq (1+\epsilon) \sum_{S \in \mathcal{S}^{opt}} c(S) \left(\sum_{t=1}^{t_S} \frac{|S^t| - |S^{t+1}|}{|S^t|} \right) \\ &\leq (1+\epsilon) \sum_{S \in \mathcal{S}^{opt}} c(S) \sum_{i=1}^{|S|} \frac{1}{i} \\ &\leq (1+\epsilon) \left(\sum_{i=1}^{\Delta} \frac{1}{i} \right) c(\mathcal{S}^{opt}) \\ &= (1+\epsilon) H(\Delta) c(\mathcal{S}^{opt}) \quad \square \end{aligned}$$

We now go back to the original facility location problem SCFLP. Set

$$\begin{aligned} w_j(L) &= \left(\lceil \sum_{i \in L} d_i/u_j \rceil f_j + \sum_{i \in L} c_{ij} \right) & (7) \\ r_j(L) &= w_j(L)/|L| & (8) \\ r_j^*(U) &= \min_{L \subseteq U} r_j(L) & (9) \end{aligned}$$

The adaptation of the set covering *Greedy* heuristic for SCFLP is described in algorithm 1. The transfer of the *Greedy* ratio $H(n)$ ($\leq 1 + \ln n$) to SCFLP depends on the approximability of subproblem (9), which can be reformulated as the following Integer Linear Program

$$\begin{aligned} \text{Minimize} \quad & \left(f_j y + \sum_{i \in U} c_{ij} x_i \right) / \left(\sum_{i \in U} x_i \right) \\ \text{s.t.} \quad & \sum_{i \in U} d_i x_i \leq u_j y \\ & \sum_{i \in U} x_i \geq 1 \\ & y \in \mathbf{N}^*, x_i \in \{0, 1\} \end{aligned}$$

When one fixes variable y as a constant, the above problem becomes a particular case of the Binary Fractional Knapsack Problem (BFKP) (see Billionnet [2] for a study of the general

Algorithm 1 /Greedy heuristic for SCFLP/
Begin
$U \leftarrow I$
Repeat
For $j \in J$ *do* find a best-possible approximation of ratio $r_j^*(U)$ of (9)
$r^*(U) = \min_{j \in J} r_j^*(U)$
Let (L^*, j^*) be the optimal pair for $r^*(U)$
$y_{j^*} := 1,\ x_{ij^*} := 1 \ \text{ for } i \in L^*$
$U \leftarrow U \setminus \{L^*\}$
Until $U = \emptyset$
output y, x
End

BFKP). For ending this section, we show that the former Integer Linear Programming problem, reformulating our subproblem (9), is NP-hard.

Proposition 3. *The problem SP of minimizing* $(fy + \sum_{1 \le i \le n} c_i x_i)/(\sum_{1 \le i \le n} x_i)$ *under the constraints* $\sum_{1 \le i \le n} x_i \ge 1$, $\sum_{1 \le i \le n} d_i x_i \le uy$, $y \in N^*$, $x_i \in \{0, 1\}$, *is NP-hard.*

Proof. We reduce the Subset-Sum Problem, known to be NP-hard [6], to SP. Given n integer numbers, $a_1, \ldots, a_n$, the Subset-Sum problem consists in deciding whether there exists a binary vector $x \in \{0, 1\}^n$ such that $\sum_{1 \le i \le n} a_i x_i = k$, given an integer number $k \neq \frac{1}{2} \sum_{1 \le i \le n} a_i$, the case $k = \frac{1}{2} \sum_{1 \le i \le n} a_i$ being known as the Partition Problem [6]. We assume that $k > \frac{1}{2} \sum_{1 \le i \le n} a_i$ without loss of generality, as otherwise we can change variables $z_i = 1 - x_i$ and look for a binary vector z such that $\sum_{1 \le i \le n} a_i z_i = \sum_{1 \le i \le n} a_i - k > \frac{1}{2} \sum_{1 \le i \le n} a_i$, turning back to the former case. We show that Subset-Sum can be formulated as a particular SP. For this, we choose in the SP instance $K = \max_i a_i$, $c_i = K - a_i$ and $d_i = a_i$ for $i = 1, \ldots, n$, $u = f = k$, and we claim that there is a Subset-Sum solution x of value k if and only if there is a SP solution (x, y) of objective value at most K.

First, let x be a Subset-Sum solution of value k. Then, in the SP instance take $y = 1$. We thus have $\sum_{1 \le i \le n} d_i x_i = \sum_{1 \le i \le n} a_i x_i = k = fy$ so the constraint is satisfied. As for the objective, its value is

$$\frac{fy + \sum_{1 \le i \le n} c_i x_i}{\sum_{1 \le i \le n} x_i} = \frac{k + K \sum_{1 \le i \le n} x_i - \sum_{1 \le i \le n} a_i x_i}{\sum_{1 \le i \le n} x_i} = K$$

Conversely, let (x, y) be a SP solution of objective value at most K, i.e., $fy + \sum_{1 \le i \le n} c_i x_i \le K \sum_{1 \le i \le} x_i$. We thus have $\sum_{1 \le i \le n} a_i x_i \ge ky$. As the inverse inequality also holds we obtain that $\sum_{1 \le i \le n} a_i x_i = ky$. As $k > \frac{1}{2} \sum_{1 \le i \le n} a_i$ then $y = 1$ and we deduce that x is indeed a Subset-Sum solution of value equal to k, which completes the proof.□

The rest of the paper is devoted to showing that the subproblem (9) of finding $r_j^*(U)$ for $j \in J$ admits a Fully Polynomial-Time Approximation Scheme (FPTAS).

4 A FPTAS for the subproblem

The algorithm for approximating optimal ratio $r_j^*(U)$ of subproblem (9) is a two-phase algorithm. In the first step, a 2-approximation of the optimal ratio is found. In the second step, costs are scaled and rounded as in the approximation algorithms of Ibarra and Kim [10] and Lawler [16] for the Knapsack Problem or the algorithm of Hassin for the Constrained Shortest Path Problem [8], and a Dynamic Programming procedure is applied. Before describing more formally the algorithm, we need to introduce the following two lemmas.

Lemma 2. *Set* $\alpha_i^j = d_i(f_j/u_j) + c_{ij}$*, and let* $S_p = \{\alpha_{i_1}^j, \ldots, \alpha_{i_p}^j\}$*, for* $p = 1, \ldots, |U|$*, be the sorted list of* p *smallest* α_i^j *values, i.e.* $\alpha_{i_l}^j \leq \alpha_{i_{l+1}}^j$ *for* $l = 1, \ldots, |U| - 1$*. Set* $S_q = arg\min_{1\leq p\leq |U|} r_j(S_p)$*. Then* $r_j(S_q)/r_j^*(U) \leq 2$*.*

Proof. We note L^* the optimal subset associated with $r_j^*(U)$ and $v(L) = \sum_{i\in L} \alpha_i^j$ for $L \subseteq U$. Then we have $v(L) \leq w(L) \leq v(L) + f_j$ for all $L \subseteq U$ (see (7) for the definition of w_j). It comes:

$$\begin{aligned}
r_j(S_q) \leq r_j(S_{|L^*|}) &= w(S_{|L^*|})/|L^*| \\
&\leq (v(S_{|L^*|}) + f_j)/|L^*| \\
&\leq (v(L^*) + f_j)/|L^*| \quad \text{as } v(S_{|L^*|}) = \min_{|L|=|L^*|} v(L) \\
&\leq (w(L^*) + f_j)/|L^*| = r_j^*(U) + f_j/|L^*| \\
&\leq 2r_j^*(U) \quad \square
\end{aligned}$$

Lemma 3. *Given* $j \in J$*, a positive real value* B*, and integer values* $\hat{f}_j$ *and* $\hat{c}_{ij}$ *for* $i \in U$*, the problem of minimizing*

$$\hat{r}_j(L) = \frac{\lceil \sum_{i\in L} d_i/u_j \rceil \hat{f}_j + \sum_{i\in L} \hat{c}_{ij}}{|L|} \tag{10}$$

over subsets $L \subseteq U$ *under the constraint* $\hat{r}_j(L) \leq B$ *can be solved in time* $O(|U|^3 B)$ *by a Dynamic Programming procedure.*

Proof. Set $U = \{i_1, \ldots, i_{|U|}\}$ and $U_l = \{i_1, \ldots, i_l\}$ for $l = 1, \ldots, |U|$. Set

$$d^*(\hat{w}, l, p) = \min_{L\subseteq U_l} \left\{ \sum_{i\in L} d_i \;\middle|\; \left(\lceil \sum_{i\in L} d_i/u_j \rceil \hat{f}_j + \sum_{i\in L} \hat{c}_{ij} \right) = \hat{w}; |L| = p \right\}$$

for $\hat{w} \in \{1, \ldots, \lfloor |U|B \rfloor\}, l \in \{1, \ldots, |U|\}, p \in \{0, \ldots, l\}$. Hence, $d^*(\hat{w}, l, p)$ is the minimum demand of a subset of U_l among all subsets of size p and (modified) cost equal to $\hat{w}$. This can be calculated by setting:

$$\begin{aligned}
d^*(\hat{w}, l, 0) &= \begin{cases} 0 \text{ if } \hat{w} = 0 \\ +\infty \text{ otherwise} \end{cases} \qquad \text{for } l = 1, \ldots, |U| \\
d^*(\hat{w}, 1, 1) &= \begin{cases} d_1 \text{ if } \hat{w} = \hat{c}_{1j} + \lceil d_1/u_j \rceil \hat{f}_j, \\ +\infty \text{ otherwise} \end{cases}
\end{aligned}$$

Algorithm 2 /FPTAS for the subproblem/
Begin
Step 1. Let $\{\alpha^j_{i_1}, \ldots, \alpha^j_{i_{|U|}}\}$ be the list of coefficients $\alpha^j_i = d_i(f_j/u_j) + c_{ij}$ sorted by non-decreasing order
$S_p := \{\alpha^j_{i_1}, \ldots, \alpha^j_{i_p}\}$ for $p = 1, \ldots, |U|$
Compute $S_q = arg\min_{1 \le p \le |U|} w(S_p)/p$
$R := r_j(S_q)$
Step 2. Set $\hat{f}_j = \lfloor f_j/(\epsilon R/4) \rfloor$ and $\hat{c}_{ij} = \lfloor c_{ij}/(\epsilon R/4) \rfloor$
Output subset L_{DP} returned by the Dynamic Programming procedure of lemma 3 with upper bound $B = 2/\epsilon$
End

and for other triples $(\hat{w}, l, p)$,

$$\begin{aligned} d^*(\hat{w}, l, p) &= \min(d^*(\hat{w}, l-1, p), \\ &\quad \left(d^*(\hat{w} - \hat{c}_{i_l j} - \lfloor d_{i_l}/u_j \rfloor \hat{f}_j, l-1, p-1) + d_{i_l}\right) z^0(\hat{w}, l, p), \\ &\quad \left(d^*(\hat{w} - \hat{c}_{i_l j} - (\lfloor d_{i_l}/u_j \rfloor + 1)\hat{f}_j, l-1, p-1) + d_{i_l}\right) z^1(\hat{w}, l, p)) \end{aligned}$$

where, for $k = 0, 1$,

$$z^k(\hat{w}, l, p) = \begin{cases} 1 \text{ if } & \lceil (d^*(\hat{w} - \hat{c}_{i_l j} - \lfloor d_{i_l}/u_j \rfloor - k, l-1, p-1) \; + d_{i_l})/u_j \rceil \\ & = \lceil d^*(\hat{w} - \hat{c}_{i_l j} - \lfloor d_{i_l}/u_j \rfloor - k, l-1, p-1)/u_j \rceil \; + \lfloor d_{i_l}/u_j \rfloor + k \\ +\infty \text{ otherwise} & \end{cases}$$

We thus look for

$$\min_{\hat{w}, p \ge 1} \{\hat{w}/p : d^*(\hat{w}, n, p) < \infty\}$$

The complexity order of this Dynamic Programming procedure is the produce of the ranges of the three integer indexes $\hat{w}$, l and p, hence the whole process runs in $O(|U|^3 B)$.□

We now introduce algorithm 2 which approximates optimal ratio $r^*_j(U)$.

Proposition 4. *Algorithm 2 is a $(1+\epsilon)$-approximation of $r^*_j(U)$ running in $O(|U|^3/\epsilon)$.*

Proof. Combining $f_j \ge (\epsilon R/4)\hat{f}_j$ and $c_{ij} \ge (\epsilon R/4)\hat{c}_{ij}$ we obtain that

$$r_j(L^*) \ge (\epsilon R/4)\hat{r}_j(L^*) \ge (\epsilon R/4) \min_{L \subseteq U} \hat{r}_j(L) = (\epsilon R/4)\hat{r}_j(L_{DP})$$

Since $r_j(L^*) \le 2R$ we get that

$$\hat{r}_j(L_{DP}) \le 2/\epsilon \tag{11}$$

which justifies that the upper bound B is set to $2/\epsilon$ in DP. Now, we have:

$$\begin{aligned}
r_j(L_{DP}) &= \frac{\lceil\sum_{i\in L_{DP}} d_i/u_j\rceil f_j + \sum_{i\in L_{DP}} c_{ij}}{|L_{DP}|} \\
&\leq \frac{\lceil\sum_{i\in L_{DP}} d_i/u_j\rceil \left(\lfloor f_j/(\epsilon R/4)\rfloor + 1\right)(\epsilon R/4) + \sum_{i\in L_{DP}} \left(\lfloor c_{ij}/(\epsilon R/4)\rfloor + 1\right)(\epsilon R/4)}{|L_{DP}|} \\
&= (\frac{\epsilon R}{4})\left(\hat{r}_j(L_{DP}) + \frac{|L_{DP}|+1}{|L_{DP}|}\right) \\
&\leq (R/2)(1+\epsilon) \qquad \text{by (11)} \\
&\leq r_j^*(U)(1+\epsilon) \qquad \text{by lemma 2}
\end{aligned}$$

The complexity of step 1 of algorithm 2 is the time of sorting coefficients α_i^j for $i \in U$, which can be done in time $|U| \ln |U|$. The complexity of step 2 is $O(|U|^3 B) = O(|U|^3/\epsilon)$. Hence, the overall complexity of algorithm 2 is $O(|U|^3/\epsilon)$. $\square$.

5 Conclusion

From propositions 1, 2 and 4 we derive the main result of the paper.

Theorem 1. *Algorithm 1 combined with FPTAS Algorithm 2 for subproblem (9) approximates SCFLP within ratio $(1+\epsilon)H(n)$ in computational time $O(mn^4/\epsilon)$.*

Since $H(n) \leq 1 + \ln n$, the gap to the inapproximability bound $\ln n$ of Feige [5] is reduced as close as possible. We can note that an adaptation of the partitioning algorithm of [8] to the SCFLP case would solve the subproblem in time $O((n^4/\epsilon)\log(n/\epsilon))$, which is significantly higher than $O(n^3/\epsilon)$. Finally, an interesting issue is whether our algorithm could improve the best-known ratio of 2 for the metric SCFLP [18]. This is quite possible since a slightly-modified version of the greedy SCP-type procedure of Hochbaum, where opening cost is set to zero once a facility is open, was proved to achieve a good approximation ratio of 1.81 for the metric UFLP [14].

References

[1] L. Alfandari and V. Paschos. Master-slave strategy and polynomial approximation. *Comp. Opt. and Applications* 16:3 (2000) 231-245.

[2] A. Billionnet. Approximation algorithms for fractional knapsack problems. *Op. Res. Letters* 30:5 (2002) 336-342.

[3] F.A. Chudak and D.P. Williamson. Improved approximation algorithms for capacitated facility location problems. In *Proc. of the 7th IFCO Conference* (1999) 99-113.

[4] V. Chvátal. A greedy-heuristic for the set covering problem. *Math. Oper. Res.* 4 (1979) 233-235.

[5] U. Feige. A threshold of $\ln n$ for approximating set cover. *J. of the ACM* 45 (1998) 634-652.

[6] M.R. Garey and D. S. Johnson., Computers and intractability. A guide to the theory of NP-completeness,W. H. Freeman, San Francisco, 1979.

[7] N. Garg, R. Khandekar and V. Pandit. Improved approximation for universal facility location. In *Proc. of* SODA (2005) 959-960.

[8] R. Hassin. Approximation schemes for the restricted shortest path problem. *Math. of Op. Res.* 17:1 (1992).

[9] D.S. Hochbaum. Heuristics for the fixed cost median problem. *Math. Prog.* 22 (1982) 148-162.

[10] O.H. Ibarra, C.E. Kim. Fast approximation algorithms for the knapsack and sum of subset problem. *J. of the ACM* 22 (1975) 463-468.

[11] K. Jain, V.V. Vazirani. Primal-dual approximation algorithms for metric facility location and k-median problems. In *Proc. of the 40th Annual IEEE Symp. on Foundations of Comp. Sc.* (1999) 2-13.

[12] K. Jain, V.V. Vazirani. Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and lagrangian relaxation. *J. of the ACM* 48 (2001) 274-296.

[13] K. Jain, M. Mahdian, A. Saberi. A new greedy approach for facility location problems. In *Proc. of the 34st Annual ACM Symp. on Th. of Computing* (2002).

[14] K. Jain, M. Mahdian, E. Markakis, A. Saberi and V.V. Vazirani. Approximation algorithms for facility location via dual fitting with factor-revealing LP. In *J. of the ACM* (2002).

[15] R.M. Karp. Reducibility among combinatorial problems. R.E Miller and J.W. Thatche Ed., Complexity of Computer Computations, 85-103, Plenum Press, NY 1972.

[16] E.L. Lawler. Fast approximation algorithms for knapsack problems. *Mathematics of Operations Research* 4 (1979) 339-356.

[17] M. Mahdian, Y. Ye and J. Zhang. Improved approximation algorithm for metric facility location problems. In *Proc. of the 5th Intl Workshop on Approximation Algorithms for Combinatorial Optimization (APPROX 2002)* (2002).

[18] M. Mahdian, Y. Ye and J. Zhang. A 2-approximation algorithm for the soft-capacitated facility location problem. In *Proc. of the 6th Intl. Workshop on Approximation Algorithms for Combinatorial Optimization* (2003) 129-140.

[19] D.B. Shmoys, E. Tardos, K. Aardal. Approximation algorithms for facility location problems. In *Proc. 29th Annual ACM Symp. on Th. of Computing* (1997) 265-274.

[20] J. Zhang, B. Chen, Y. Ye. Multi-exchange local search algorithm for capacitated facility location problem. In *Proc.* IPCO (2004) 219-233.